2007

# Group Rekeying Schemes for Secure Group Communication in Wireless Sensor Networks

Yong Wang
*University of Nebraska-Lincoln*, ywang@cse.unl.edu

Byrav Ramamurthy
*University of Nebraska-Lincoln*, bramamurthy2@unl.edu

*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.*
ICC '07. IEEE International Conference on Communications, 2007.
Digital Object Identifier: 10.1109/ICC.2007.566 Publication Year: 2007 , Page(s): 3419 - 3424

# Group Rekeying Schemes for Secure Group Communication in Wireless Sensor Networks

Yong Wang and Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115 USA
{ywang, byrav}@cse.unl.edu

***Abstract*— Wireless sensor networks are promising solutions for many applications. However, wireless sensor nodes suffer from many constraints such as low computation capability, small memory, limited energy resources, and so on. *Grouping* is an important technique to localize computation and reduce communication overhead in wireless sensor networks. In this paper, we use grouping to refer to the process of combining a set of sensor nodes with similar properties. We propose two centralized group rekeying (CGK) schemes for secure group communication in sensor networks. The lifetime of a group is divided into three phases, i.e., group formation, group maintenance, and group dissolution. We demonstrate how to set up the group and establish the group key in each phase. Our analysis shows that the proposed two schemes are computationally efficient and secure.**

## I. Introduction

Wireless Sensor Networks (WSNs) are used in many applications in military, environmental and health related areas. However, nodes in a WSN suffer from many constraints such as low computation capability, small memory, limited energy resources, and so on. Grouping is an important technique to localize computation and reduce communication overhead in WSNs.

The most common method of grouping is clustering. The essential operation in sensor node clustering is to select a set of cluster heads among the sensors in the network, and cluster the rest of the nodes with these heads [1]. Cluster heads are responsible for coordination among the nodes within their clusters, and communication with each other and/or with external observers on behalf of their clusters. Many routing protocols and key management protocols have been proposed using the clustering technique [2], [3].

However, grouping goes far beyond clustering. In this paper, we use grouping to refer to the process of combining a set of sensors with similar properties. The essential operation in sensor node grouping is to dynamically combine a set of sensors based on the observed events. The result of the grouping is a group. Unlike clustering focusing on the whole sensor network, grouping is only involved with sensors in a small region. Without additional clarifications, the term grouping in this paper refers to the local combination of a set of sensor nodes. There are many similarities between clustering and grouping, for example:

- Sensors in a cluster or a group are usually geographically close to each other.
- Both clustering and grouping are used to localize computation and reduce communication overhead.
- A cluster usually has a cluster head and a group may have a group controller.

However, important differences exist between clustering and grouping. The main differences are listed below:

- Clustering is a global concept while grouping usually focuses on a small region. When clustering is used in a sensor network, the whole sensor network is divided into clusters. However, grouping usually involves with a relatively small number of sensors. These sensors are combined together based on the defined properties.
- Clustering and grouping could be adopted separately or together. They do not depend on each other. Grouping can be carried out with clustering or without.
- When clustering and grouping are both used to organize a sensor network, a group could be a part of a cluster, or even the union of several clusters.
- Clusters are decided by the partition algorithm adopted in the sensor networks. There is no relation between the clusters and the observed events. However, groups are usually activated by events. A group is set up and dissolved on the fly.

Security is an important research area in sensor networks [4]. In this paper, we focus on secure group communication (SGC) [5]. Secure group communication in sensor networks refers to a scenario in which sensors in a group can send and receive messages to/from group members in a way that outsiders are unable to glean any information even when they are able to intercept the messages. Secure group communication depends on the *group key* to protect the messages. The security requirements of group communication include authentication, confidentiality, integration, freshness etc. [5]. In addition, secure group communication also requires forward secrecy and backward secrecy [5]. The obvious benefit of secure group communication to WSNs is that outside nodes are unable to obtain any messages transmitted to the group. Recent research has also revealed that the group key can be used for filtering out false data injected in the sensor networks [6], [7].

Although a few papers [8], [9] discussed secure group communication in sensor networks in the literature, the problem has not been well-studied. Previous works on secure group communication either consider the whole sensor network as

**1-4244-0353-7/07/$25.00 ©2007 IEEE**

a group or define the immediate neighboring nodes around a sensor as a group. However, grouping is more general than these two cases.

Our contributions in this paper are as follows: We formally define the grouping and secure group communication problem in WSNs. We differentiate between the concepts of clustering and grouping. We propose two centralized group rekeying schemes for secure group communication in WSNs and further evaluate their performances in various group settings.

The paper is organized as follows: Section II discusses the related work. Section III introduces grouping and its properties. Section IV presents our proposed centralized group rekeying schemes, followed by the security and performance analysis in Section V, the simulation and results in Section VI. Section VII concludes the paper.

## II. Related work

The secure group communication problem has been extensively studied in the context of secure multicast in wired or wireless networks. Many centralized solutions and a few distributed solutions have been proposed. However, most of them are not suitable for WSNs. For example, the centralized schemes proposed in [10], [11] assume a key tree is maintained in the central controller. However, none of them considers the management overhead of such a key tree structure in the central controller, which is important in sensor networks due to the constraints on the sensor nodes. The distributed schemes, such as [12], [13], require excessive computation (exponential operations) to generate and update the group key, which are also unbearable in WSNs.

A few papers [8], [9], [14], [15] address the secure group communication problem in the context of sensor networks. The authors in [8] proposed a scheme using a key tree to manage group members as they join or leave the group. However, the authors did not provide the details of the group rekeying process. In [9], the authors proposed an energy-efficient level-based hierarchical system for sensor networks which also includes a group key management scheme. The proposed group rekeying scheme requires many exponential operations which makes it possibly not practical in sensor networks. In [14], the authors proposed a centralized group rekeying scheme based on logical key tree hierarchy for WSNs. In all these three works [8], [9], [14], the base station is regarded as the central controller and the whole sensor network is considered as a group. In [15], the authors proposed a group rekeying scheme for filtering false data in sensor networks. The group is defined as the immediate neighboring nodes around a sensor in the scheme. However, the authors did not address the group rekeying problem when the group includes sensor nodes separated by multiple hops.

In the following sections we present our proposed centralized group rekeying schemes. We use the following notation for the remainder of this paper:

- $A, B$ are principals such as communicating nodes.
- $ID_A$ denotes the sensor identifier of node $A$.
- $e(A,T)$ is a set of events observed by sensor $A$ in time period $T$.
- $K_{AB}$ denotes the secret pairwise key shared between $A$ and $B$.
- $M_K$ is the encryption of message $M$ with key $K$.
- $MAC(K,M)$ denotes the computation of the message authentication code of message $M$ with key $K$.
- $A \longrightarrow B$ denotes $A$ unicasts a message to $B$.
- $A \longrightarrow *$ denotes $A$ broadcasts a message to its neighbors.

## III. Grouping and its properties

As we discussed before, grouping refers to the process of combining a set of sensors with similar properties. These properties usually refer to the events observed by the sensors. A group can be defined by many aspects. For example, all photo sensors activated in the last one minute form a group; the temperature sensors with temperature more than $100^{\circ}\mathrm{C}$ form a group. Without loss of generality, we define a group $G$ as a set of sensors $A$ in region $R$ which observe an event $E$ in a period of time $T$:

$$G = \{A | E \in e(A,T) \ \ and \ A \ in \ R\} \tag{1}$$

The lifetime of a group can be divided into three phases, i.e., *group formation*, *group maintenance*, and *group dissolution*.

In the *group formation* phase, the sensor nodes which satisfy the defined criteria form a group. The process of group formation is usually triggered by a special node, which is called a *group controller*. The group controller can be decided by the *controller selection process*. A simple way to decide a group controller is as follows: when an event $E$ occurs in the field, the sensor detecting this event and having the strongest signal stands out as the group controller. The group formation phase is ended with all the group members receiving the group key. Then, the group maintenance phase begins.

The *group maintenance phase* is divided into sessions. The duration of sessions can be fixed or dynamic depending on the applications. The group controller is responsible for distributing the group key to the sensor nodes at each session. When new sensors join a group or existing members leave the group, the group membership must be updated. In addition, when a compromised group member is detected, the compromised group member must also be removed from the group. Since the group key is updated during each session, the leaving members and the compromised members will not obtain the new group key during the next session.

In the *group dissolution* phase, the sensor nodes in the group are not bound together anymore. The key materials set up before should be released.

## IV. Group rekeying schemes for SGC in WSNs

In this section, we present two centralized group rekeying (CGK) schemes for secure group communication in WSNs. We assume that there is a *secure channel* between the sensor node and the base station. By a *secure channel*, we mean a channel that offers confidentiality, data authentication, integrity, and freshness. The key materials to build the secure

channel can be set up by the key management protocols described in [3], [16].

### A. Scheme 1

Scheme 1 is based on Blundo's theory [17]. The group key is distributed to the group members through unicasting. The group formation phase in Scheme 1 is described below.

1) Setup: Before sensor nodes are distributed, the setup server randomly generates a bivariate $t$-degree polynomial $f(x,y) = \sum_{i,j=0}^{t} a_{ij}x^i y^j$ over a finite field $F_q$ where $q$ is a prime number that is large enough to accommodate a cryptographic key such that it has the property that $f(x,y) = f(y,x)$. For each sensor *i*, the setup server computes a polynomial share of $f(x,y)$, that is, $f(i,y)$, and loads the single-variate polynomial $f(i,y)$ to the sensor *i*. For any two sensor nodes *i* and *j*, node *i* can compute the common key $f(i,j)$ by evaluating $f(i,y)$ at point *j*, and node *j* can compute the same key $f(j,i) = f(i,j)$ by evaluating $f(j,y)$ at point *i*.
2) Broadcast interest: Once the group controller is identified, it first obtains a group identifier $gid$ from the base station and then generates a random key $K_g$ as the group key. Subsequently, the group controller broadcasts a message requesting expression of interest in a particular event $E$ to its neighboring nodes which are reachable in at most *L* hops (global broadcasting is not necessary):

$$I \longrightarrow * : ID_I|gid|E$$

3) Join: All the receivers observing the same event $E$ send a join request to the group controller *I*:

$$A \longrightarrow I : ID_A|gid|E, MAC(K_{AI}, ID_A|gid|E)$$

where $K_{AI}$ is the pairwise key shared by the group controller *I* with the sensor *A*. During the period when interest and join messages are transmitted in the network, route tables are set up in the en-route nodes and the group controller.
4) Group key distribution: Once the group controller authenticates the join request, the group controller unicasts the group key $K_g$ encrypted by the pairwise key to the sensor *A*:

$$I \longrightarrow A : \{K_g\}_{K_{AI}}$$

In the group key maintenance phase, the group controller keeps track of the join and leave requests in the group and repeats step 4 to update the group key during each session. The group controller maintains a table to keep the latest membership. Without receiving the group key update messages in a period of time $\tau_s$, the key materials become obsolete and the group members can start the group dissolution process.

Let $|G| = n$. Scheme 1 requires one local broadcast in the group formation phase. The group controller may receive $n$ join requests and needs to send the group key to $n$ members. Thus, to set up the group key among $n$ members, it requires $2n$ unicasts and one local broadcast. To update the group key, it requires $n$ unicasts of messages.

Note that Scheme 1 requires $n$ unicasts of messages to update the group key which may cause heavy traffic in the area when the group size is large. We propose Scheme 2 which uses local broadcast to replace the unicasts to reduces the communication overhead when updating the group key.

### B. Scheme 2

Scheme 2 is based on Blundo's theory [17] and the personal key share distribution scheme [18]. The group key in Scheme 2 is distributed through broadcasting. The group formation phase in Scheme 2 is described below. The setup, broadcast interest and join steps (1, 2, and 3) are the same as in Scheme 1 and are omitted.

4) Secret share distribution: The group controller randomly picks a $2t$-degree masking polynomial, $h(x) = h_0 + h_1 x + \cdots + h_{2t}x^{2t}$, over $F_q$. Each group member $A_i$ gets the personal secret, $S_i = h(i)$, from the group controller via the secure communication channel between them:

$$I \longrightarrow A_i : \{S_i\}_{K_{AI}}$$

5) Distinct share broadcast: Given a set of IDs of revoked group members, $R = \{r_1, r_2, \cdots, r_w\}$, $w \leq t$, the group controller randomly picks a $t$-degree polynomial $p(x)$ and constructs $q(x) = K_g - p(x)$. Then, the group controller distributes the shares of the $t$-degree polynomials $p(x)$ and $q(x)$ to non-revoked sensors using the following broadcast message:

$$\begin{aligned} B &= \{R\} \\ &\cup \{P(x) = g(x)p(x) + h(x)\} \\ &\cup \{Q(x) = g(x)q(x) + h(x)\} \end{aligned}$$

where the revocation polynomial $g(x)$ is constructed as $g(x) = (x - r_1)(x - r_2)\cdots(x - r_w)$.
6) Group key recovery: If any non-revoked sensor node $A_i$ receives such a broadcast message, it evaluates the polynomial $P(x)$ and $Q(x)$ at point $i$ and gets $P(i) = g(i)p(i) + h(i)$ and $Q(i) = g(i)q(i) + h(i)$. Because $A_i$ knows $h(i)$ and $g(i) \neq 0$, it can compute $p(i) = \frac{P(i)-h(i)}{g(i)}$ and $q(i) = \frac{Q(i)-h(i)}{g_j(i)}$. $A_i$ can finally compute the new group key $K_g = p(i) + q(i)$. The revoked sensors cannot recover the group key because $g(x) = 0$.

In the group maintenance phase, the group controller repeats steps 5 and 6 to distribute the group key during each session. Similar to Scheme 1, Scheme 2 requires $2n$ unicasts and one local broadcast to set up the group key among $n$ members. However, to update the group key, it only requires one broadcast of messages.

### C. Broadcast authentication

A missing link in both of the above schemes is how a group controller broadcasts local authenticated messages (messages requesting expression of interest in a particular event). In the absence of authentication of broadcast messages, an adversary

can impersonate a group controller and start a group. The scheme in [19] can be adopted for broadcasting authenticated messages.

The authors in [19] proposed a practical broadcast authentication scheme which supports multi-senders in a WSN. The scheme can be used here to enable a sensor to broadcast local authenticated messages. Unlike the key materials for broadcasting authentication messages required to be loaded on the senders in the pre-distribution stage in [19], we require those key materials to be sent to the group controller dynamically in the group setup stage. When a group controller is identified in the group formation phase, the group controller needs to send a request to the base station to obtain a group identifier. At that time, the base station can also load the required materials for authenticating broadcast messages on the group controller using a secure channel. Thus, the group controller can use the obtained key materials to broadcast authenticated messages. The adversaries cannot impersonate the group controller because they cannot authenticate themselves to the base station. (For more detailed information about the practical broadcast authentication scheme, please refer to [19].)

Note that our group rekeying schemes use an adapted broadcast authentication scheme from [19]. Thus, the sensor network should be loosely time synchronized to meet the requirements in [19].

## V. Security and performance analysis

In Section IV, we noted that the group controller has to be authenticated by the base station to get proper materials to broadcast authenticated messages. Thus, an adversary cannot impersonate a group controller to start the group formation process.

Further, according to the scheme in [19], the group controller is only granted the ability to broadcast messages in some specific time intervals. After these assigned intervals, the group controller cannot broadcast messages to the sensor network anymore. Therefore, even if the group controller is compromised, the adversaries cannot utilize the group controller to indefinitely broadcast messages to the whole sensor network.

In the group formation process, the group controller also authenticates the joining members to ensure that only the qualified sensors can join the group.

In case a sensor is compromised, the adversary can know the group keys which it possesses but cannot obtain the group keys not available to the sensor. Once the compromised sensor is detected by some intrusion detection techniques such as [20], [21], the compromised sensor could be removed from the group.

The pairwise key shared by the group controller with each joining member is built using Blundo's theory [17]. To set up the pairwise key, the sensor node needs to evaluate the polynomial value at point $(i, j)$. Thus, the additional computation overhead for calculating the pairwise key is almost negligible. To use Blundo's theory, each sensor node *i* needs to store a *t*-degree polynomial $f(i, x)$, which occupies $(t + 1) \log q$ storage space. In addition, the group controller also needs $n$ storage units for the pairwise keys shared with the group members and one unit for the group key. To use the scheme in [19], the sensor nodes need to be loaded with some pre-distributed values. The storage requirements for broadcasting authenticated messages are the same as the scheme in [19]. The two proposed schemes are compared in Table I.

TABLE I

Comparison between Scheme 1 and Scheme 2.

| | Scheme 1 | | Scheme 2 | |
|---|---|---|---|---|
| message | nums | size | nums | size |
| Broadcast requiring msg | 1 | $O(\log q)$ | 1 | $O(\log q)$ |
| Join request | $n$ | $O(\log q)$ | $n$ | $O(\log q)$ |
| Secret share distribution | $n/a$ | $n/a$ | $n$ | $O(\log q)$ |
| Group key distribution | $n$ | $O(\log q)$ | 1 | $O(t \log q)$ |
| Group key update | $n$ | $O(\log q)$ | 1 | $O(t \log q)$ |
| Group key revocation | $n$ | $O(\log q)$ | 1 | $O(t \log q)$ |

## VI. Simulation and results

The performance of the two proposed schemes was evaluated in SENSIM [22], a component-based discrete-event simulator for sensor networks. Each sensor node in SENSIM consists of six components, i.e., app, net, mac, phy, event generator, and battery. The two proposed schemes are implemented in the network component independently. In the simulation, all the packets sent to the MAC layer are guaranteed to be received at the receivers. Thus, no packet collisions are considered and the performance evaluated in the simulation is under ideal conditions.

We consider both Scheme 1 and Scheme 2 operating on a finite field $F_q$, where $q$ is a 56-bit integer. The polynomial degree $t$ in Blundo's theory is set to $t = 4$ which gives the message size of Scheme 1 as 8 bytes and the max message size of Scheme 2 as 136 bytes. We use the simulator parameters that represent the Mica2 Mote radio characteristics. These parameters are shown in Table II.

TABLE II

Characteristic data for the Mica2 sensor platform.

| Field | Value |
|---|---|
| Effective data rate | 19.2kbps |
| Transmit power | 36mW |
| Receive power | 14.4mW |
| Idle power | 14.4mW |
| Sleep | 0.015mW |
| Transition power | 28.8mW |
| Transition time | 800$\mu$s |

We assume that 1000 nodes are uniformly dispersed in a field with dimension $2000m \times 2000m$ and we set the group controller at $(1088, 1151)$. The evaluation metrics include the group formation time, the group key update time, the energy consumption in group controller, and the energy consumption in group member nodes. The group formation time is the time duration from the group controller broadcasting the interest message till all the group members receive the first group key. We test the two schemes for different group sizes. The group size is decided by a maximum count (max hops) along

the routes in which the interest message is forwarded and we assume that all sensor nodes which hear the message become group members. For each group size, we run the simulation ten times and the average value is measured. Table III shows the group size and the max hops in our simulation.

TABLE III

GROUP SIZE AND THE MAX HOPS IN THE SIMULATION.

| L (max hops) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Group size | 16 | 38 | 70 | 126 | 206 | 284 | 389 | 503 |

Figure 1 shows the group formation time as the number of max hops increases. It shows that Scheme 2 requires more time to set up the group because additional transmission of key materials is required. Further, when the number of max hops is greater than three, it takes a long time ($> 1\ min$) for these two schemes to set up the group. It indicates that the number of max hops on routes which the interest messages are allowed to traverse should be less than four. Figure 2 shows the group key update time in the group maintenance phase. Scheme 2 is far better than Scheme 1 due to the use of broadcasting to replace unicasting when updating the group key.

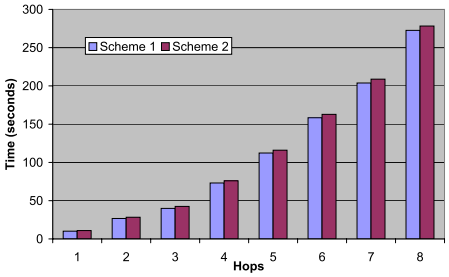


Fig. 1. Group formation time: Scheme 2 requires more time to set up the group because of additional transmission of key materials is required.

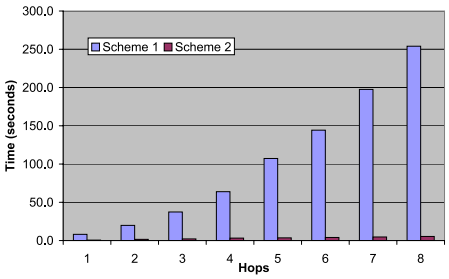


Fig. 2. Group key update time: Scheme 2 is far better than Scheme 1 in the stage of group maintenance.

Figures 3 and 4 show the average group controller energy consumption in the group formation and group key update phrases. As the figures indicate, although Scheme 1 requires less energy for the group controller to set up the group, the group controller in Scheme 1 consumes much more energy to update the group key. Because the group key is updated at regular time intervals, Scheme 1 may cause the group controller to deplete its energy much faster than Scheme 2.

Figures 5 and 6 show the average group member energy consumption in the group formation and group key update

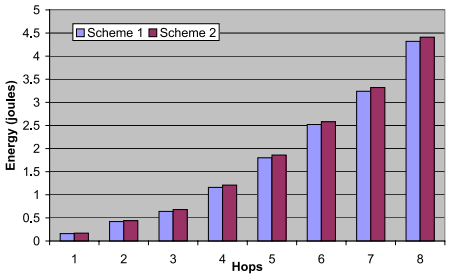


Fig. 3. Average group controller energy consumption: group formation phase. Scheme 1 requires less energy to set up the group.

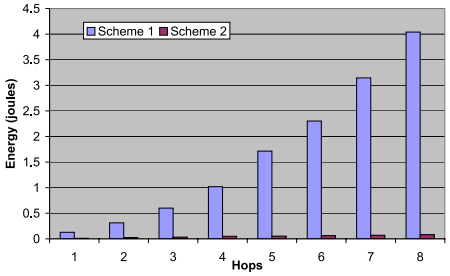


Fig. 4. Average group controller energy consumption: group key update phase. Scheme 2 is far better than Scheme 1 when updating the group key.

phases. As the figures show, Scheme 1 is slightly better than Scheme 2 in the group formation phase but Scheme 2 is far better than Scheme 1 in the group key update phase. In consideration of the group key is updated at regular time intervals, Scheme 2 is much better than Scheme 1 in the energy consumption in group member sensor nodes.

Figures 7 and 8 show the energy distribution among group members when the number of max hops is three. As the figures show, Scheme 1 may cause the energy to be distributed unevenly in the group formation phase. However, the energy is distributed more evenly in Scheme 2 in both the group formation and group update phases.

To summarize, with respect to the group formation time and the energy consumption in the group controller and the group member sensor nodes, Scheme 1 is slightly better than Scheme 2 in the group formation phase; however, Scheme 2 is far better than Scheme 1 in the group key update phase. Because the group key is updated at regular time intervals, Scheme 2 is better than Scheme 1 for secure group communication in the sensor networks.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two centralized group rekeying (CGK) schemes for secure group communication in WSNs. Our analysis shows that both the schemes are efficient in computation and secure in the sense of group communication. Simulation results also show that Scheme 2 is a better option than Scheme 1 to be a group rekeying scheme for secure group communication in WSNs.

As the simulation shows, Scheme 2 is scalable to large groups in the group maintenance phase. However, the group formation phase may take a long time ($> 2\ mins$) when the number of max hops is great than four. The group formation phase needs to be improved.

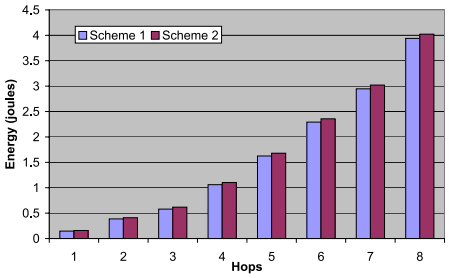


Fig. 5. Average group member energy consumption: group formation phase. The group members in Scheme 1 require less energy to set up the group.

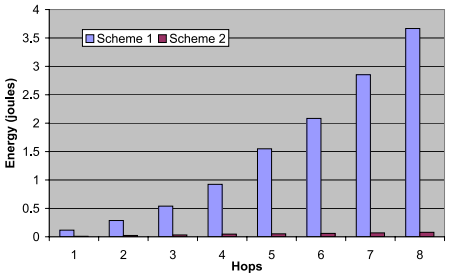


Fig. 6. Average group member energy consumption: group key update phase. Scheme 2 is far better than Scheme 1 when updating the group keys.

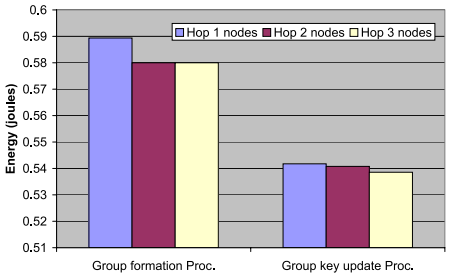


Fig. 7. Average group member energy distribution by hops: Scheme 1, max hops=3. Scheme 1 may cause the energy distributed unevenly in the group formation phase.

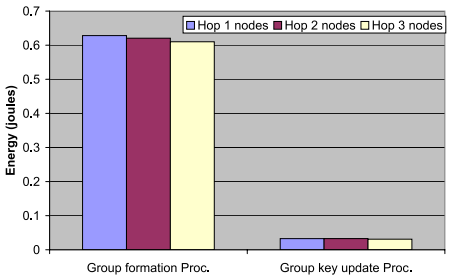


Fig. 8. Average group member energy distribution by hops: Scheme 2, max hops=3. In Scheme 2, the energy is distributed more evenly than Scheme 1 in the group formation and group update phases.

## ACKNOWLEDGEMENTS

This work is partially supported by NSF Grant No. CCR-0311577.

## REFERENCES

[1] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *Proceedings of IEEE INFOCOM*, Hongkong, 2004, pp. 629–640.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of 33rd Annual Hawaii International Conference on System Sciences*, January 2000.

[3] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2003, pp. 62–72.

[4] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 2, 2006.

[5] X. Zou, B. Ramamurthy, and S. S. Magliveras, *Secure Group Communications Over Data Networks*. Springer, 2005.

[6] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proceedings of IEEE INFOCOM*, Hongkong, 2004.

[7] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2004, pp. 259–271.

[8] N. Thepvilojanapong, Y. Tobe, and K. Sezaki, "A proposal of secure group communication for wireless sensor networks," in *The 23th Computer Security (CSEC) Group Meeting, IPSJ*, Tokyo, Japan, Dec. 2003, pp. 47–52.

[9] M. Tubaishat, J. Yin, B. Panja, and S. Madria, "A secure hierarchical model for sensor network," *SIGMOD Rec.*, vol. 33, no. 1, pp. 7–13, 2004.

[10] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, no. 1, pp. 16–30, 2000.

[11] D. Balenson, D. McGrew, and A. Sherman, "Key management for large dynamic groups: One-way function trees and amortized initialization," IETF Internet draft, August 2000.

[12] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in *CCS '96: Proceedings of the 3rd ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 1996, pp. 31–37.

[13] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2000, pp. 235–244.

[14] R. D. Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. J. M. Havinga, "LKHW: A directed Diffusion-Based secure multicast scheme for wireless sensor networks," in *ICPPW '03: Proceedings of the 32nd International Conference on Parallel Processing Workshops*. IEEE Computer Society Press, 2003, pp. 397–406.

[15] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: a predistribution and local collaboration-based approach," in *Proceedings of IEEE INFOCOM*, Miami, March 13-17 2005.

[16] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2002, pp. 41–47.

[17] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1993, pp. 471–486.

[18] D. Liu, P. Ning, and K. Sun, "Efficient self-healing group key distribution with revocation capability," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2003, pp. 231–240.

[19] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," in *MobiQuitous '05: Proceedings of The 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, July 2005, pp. 118–129.

[20] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 255–265.

[21] G. Wang, W. Zhang, C. Cao, and T. L. Porta, "On supporting distributed collaboration in sensor networks," in *Proceedings of MILCOM*, 2003.

[22] Y. Wang and B. Ramamurthy, "SENSIM: SEnsor Network SIMulator (Version 0.1)," August 2006. [Online]. Available: http://cse.unl.edu/∼ywang/sensim.htm